Claim 1 (currently amended): A method for designing a software architecture for utilizing software components in building N-tier software applications, the method comprising:

a) specifying a set of software component rules for developing software components;

b) specifying a set of tier rules for developing a plurality of tiers wherein each tier comprises a plurality of software components and performs a predetermined function, each software component comprising a software object, the tier rules further comprising:

i) a set of association rules by which at least one software component developed using the software component rules is associated with or disassociated from at least one tier developed with the set of tier rules;

ii) a set of tier framework rules to provide an architected context for software components associated with a tier; and

iii) a set of package rules to provide for logical grouping of interfaces within a framework defined by the tier framework rules to provide a set of specific behaviors for the tier; and

c) specifying a set of assembly rules~~, the assembly rules comprising association rules~~ by which each tier is associated ~~with at least one other tier and linkage rules by which each tier is~~ and linked to at least one other tier.

Claim 18 (currently amended): A method for generating software components for use in an N-tier software application, the software components having a predetermined structure, the method comprising:

providing a software component architecture comprising a plurality of tiers, wherein each tier comprises a plurality of software components and performs a predetermined function, each software component comprising a software object, each tier further comprising a predetermined set of interfaces for that tier, the interfaces defining a set of functionality capable within that tier; ~~and~~

specifying a set of association rules by which at least one software component developed using software component rules is associated with or disassociated from at least one tier developed with a set of tier rules;

specifying a set of tier framework rules to provide an architected context for software components associated with a tier;

specifying a set of package rules to provide for logical grouping of interfaces within a framework defined by the tier framework rules to provide a set of specific behaviors for the tier;

specifying a set of assembly rules by which each tier is associated and linked to at least one other tier; and

wherein at least one of the software components in a selected one of the plurality of tiers provides a predetermined set of interfaces.

Claim 20 (cancelled).

Claim 21 (currently amended): The method of claim ~~20~~ 18 wherein the tier framework rules further comprise:

specifying rules for determining dependencies between a framework and at least one other framework;

specifying how properties and interfaces are grouped in the framework; and

specifying the interfaces to be used in the framework;.

Claim 23 (currently amended): A method of system design for an N-tier architecture, the architecture comprising software components and tiers, the method comprising:

determining a set of application requirements;

determining a list of ~~required models and~~ software components to satisfy the application requirements, wherein each software component comprises a software object;

logically grouping the software components into extensible tiers, the tiers having a set of tier rules, the tier rules comprising a set of association rules by which at least one software component developed using software component rules is associated with or disassociated from at least one tier developed with the set of tier rules, the tier rules further comprising a set of tier framework rules to provide an architected context for software components associated with a tier, the tier rules further comprising a set of package rules to provide for logical grouping of interfaces within a framework defined by the tier framework rules to provide a set of specific behaviors for the tier;

specifying a set of assembly rules by which each tier is associated and linked to at least one other tier

~~determining if each software component in each tier is available in an inventory of components;~~

~~using each software component found in the inventory if that software component is a required software component;~~

~~restructuring a portion of the software components in the inventory;~~

~~adding additional software components if no existing software component in the inventory satisfies a requirement or is modifiable to satisfy the requirement;~~

~~associating a plurality of software components with each required tier wherein each software component comprises a software object; and~~

~~developing an application by defining and implementing linkages between the required tiers.~~

Claim 29 (currently amended): A system for designing a software architecture for use in generating software components for building software applications, the system comprising:

at least one processing unit;

at least one memory store operatively connected to the processing unit;

N-tier design software executable within the at least one processing unit, wherein each tier comprises a plurality of software components and performs a predetermined function, each software component comprising a software object;

software architecture specifications resident in the memory store for use by the N-tier design software, the software architecture specifications comprising specifications for a set of software component rules for developing software components, specifications of a set of tier rules for developing tiers, and specifications of a set of assembly rules, the tier rules comprising a set of association rules by which at least one software component developed using software component rules is associated with or disassociated from at least one tier developed with the set of tier rules, the tier rules further comprising a set of tier framework rules to provide an architected context for

software components associated with a tier, the tier rules further comprising a set of package rules to provide for logical grouping of interfaces within a framework defined by the tier framework rules to provide a set of specific behaviors for the tier, the set of assembly rules associating and linking each tier with at least one other tier;

an input device, operatively in communication with the processing unit, for permitting input of the software architecture specifications;

an output device, operatively in communication with the processing unit; and

a communications pathway operatively connected to the processing unit.

Claim 34 (currently amended): An N-tier software architecture stored in a storage media, the storage media comprising:

a first plurality of binary values for developing software components using software component rules;

a second plurality of binary values for developing a plurality of tiers using tier rules wherein each tier comprises a plurality of software components and performs a predetermined function, each software component comprising a software object, the tier rules comprising a set of association rules by which at least one software component developed using software component rules is associated with or disassociated from at least one tier developed with the set of tier rules, the tier rules further comprising a set of tier framework rules to provide an architected context for software components associated with a tier, the tier rules further comprising a set of package rules to provide for logical grouping of interfaces within a framework defined by the tier framework rules to provide a set of specific behaviors for the tier; and

a third plurality of binary values for assembling software applications from tiers and software components, the third plurality of binary values associating and linking each tier with at least one other tier.

Claim 35 (currently amended): An article of manufacture, comprising:

a computer storage medium having a computer program encoded therein for designing a software architecture for utilizing software components in building multiple-tier software applications, the computer storage medium including:

code for specifying a set of software component rules for developing software components wherein each software component comprises a software object;

code for specifying a set of tier rules for developing a plurality of tiers wherein each tier comprises a plurality of software components and performs a predetermined function, the tier rules comprising a set of association rules by which at least one software component developed using software component rules is associated with or disassociated from at least one tier developed with the set of tier rules, the tier rules further comprising a set of tier framework rules to provide an architected context for software components associated with a tier, the tier rules further comprising a set of package rules to provide for logical grouping of interfaces within a framework defined by the tier framework rules to provide a set of specific behaviors for the tier; and

code for specifying a set of assembly rules ~~having association rules~~ by which each tier is associated ~~with at least one other tier and linkage rules by which each tier is~~ and linked to at least one other tier.